Listing of Claims

This listing of claims replaces all prior versions, and listings, of claims in the application:

1. (Currently Amended) An execution unit for execution of multiple context threads, comprising ~~comprises~~:

an arithmetic logic unit to process data for executing threads;

control logic to control the operation of the arithmetic logic unit; and

a general purpose register set to store and obtain operands for the arithmetic logic unit, the register set ~~constructed with~~ comprising a plurality of two-ported random access memory devices ~~architecture, with the register set divided into a plurality of banks~~, ~~each bank having no more than~~ the register set comprising two effective read ports and one effective write port, each bank ~~and~~ being capable of performing a read and a write to two different words ~~with the two ports~~ in the same processor cycle.

2. (Original) The execution unit of claim 1 wherein the register set is logically partitioned into a plurality of relatively addressable windows.

3. (Currently Amended) The execution unit of claim 2 wherein the number of windows of the register set is ~~according~~ related to the number of threads that ~~can~~ execute in the processor.

4. (Currently Amended) The execution unit of claim 1 ~~where~~ wherein ~~the~~ relative addressing allows ~~the currently~~ an executing thread to access ~~to any of~~ the ~~registers~~ register set relative to the starting point of a window ~~of registers~~.

5. (Currently Amended) The execution unit of claim 1 wherein the register set is absolutely addressable, where ~~any one of~~ the ~~addressable registers~~ register set may be accessed ~~by~~ for ~~the currently~~ an executing thread by providing ~~the~~ an exact address ~~of the register~~.

6. (Currently Amended) The execution unit of claim 1 wherein the control logic further comprises:

context event switching logic fed by signals from a plurality of shared resources, ~~with~~ the signals causing the context event switching logic to indicate that threads are either available or unavailable for execution.

7. (Currently Amended) The execution unit of claim 6 wherein the control logic addresses a first set of memory locations for storing a list of available threads that correspond to threads that are ready to be executed and a second set of memory locations for storing a list of unavailable threads that are not ready to be executed.

Claims 8-14 (Canceled)

15. (Currently Amended) A method for executing multiple context threads, comprising ~~comprises~~:

processing data for executing threads within an arithmetic logic unit;

operating control logic to control the arithmetic logic unit;

storing and obtaining operands for the arithmetic logic unit within a general purpose register ~~with the register~~ set ~~constructed with~~ comprising a plurality of two-ported random access memory ~~architecture~~ devices~~; and~~

~~arranging the register set into a plurality of banks~~, ~~each bank having no more than~~ the register set comprising two effective read ports and one effective write port and each bank being capable of performing a read and a write to two different words ~~with the two ports~~ in the same processor cycle.

16. (Currently Amended) The method of claim 15 wherein the register set is relatively addressable.

17. (Currently Amended) The method of claim 15 further comprising:

arranging the register set into a number of windows according to the number of threads that ~~can~~ execute ~~in the processor~~.

18. (Currently Amended) The method of claim 17 wherein storing and obtaining further comprises:

addressing the ~~registers~~ register set ~~by~~ for ~~the currently~~ an executing thread by providing a relative register address that is relative to the starting point of a window ~~of registers~~.

19. (Currently Amended) A processor unit comprising ~~comprises~~:

an execution unit for execution of multiple context threads, the execution unit comprising:

an arithmetic logic unit to process data for executing threads;

control logic to control the operation of the arithmetic logic unit; and a general purpose register set to store and obtain operands for the arithmetic logic unit, the register set ~~constructed with~~ comprising a plurality of two-ported random access memory ~~architecture~~ devices, ~~the register set being arranged into a plurality of banks, each bank having no more than~~ the register set comprising two effective read ports and one effective write port.

20. (Currently Amended) The processor of claim 19 wherein the register set is logically partitioned into a plurality of relatively addressable windows, where the number of windows of the register set is ~~according~~ related to the number of threads that ~~can~~ execute in the processor.

21. (Currently Amended) The processor of claim 20 ~~where~~ wherein ~~the~~ relative addressing allows ~~the currently~~ an executing thread to access ~~to any of~~ the ~~registers~~ register set relative to the starting point of a window ~~of registers~~.

22. (Currently Amended) The processor of claim 20 wherein the register set is absolutely addressable, where ~~any one of~~ the ~~addressable registers~~ register set may be accessed ~~by~~ for ~~the currently~~ an executing thread by providing ~~the~~ an exact address ~~of the register~~.

23. (Currently Amended) The processor of claim 20 further comprising ~~comprises~~:

a set of memory locations for storing a list of available threads that ~~correspond to threads that~~ are ready to be executed;

a set of memory locations for storing a list of unavailable threads that are not ready to be executed; and

context event switching logic fed by signals from a plurality of shared resources, ~~with~~ the signals causing the context event switching logic to indicate which threads are either available or unavailable for execution.

24. (Currently Amended) The processor of claim 23 wherein execution of a context swap instruction causes a currently running thread to be swapped out to the list of unavailable ~~thread memory set~~ threads and a thread from the list of available ~~thread memory set~~ threads to begin execution within a single execution cycle.

25. (Currently Amended) The processor of claim 23 wherein execution of a context swap instruction specifies one of the ~~signal inputs~~ signals and upon receipt of the specified signal

~~input~~ causes ~~the~~ a swapped out thread to be stored in the available thread memory set.

26. (Currently Amended) The processor of claim 23 wherein execution of a context swap instruction specifies a defer_one operation which causes execution of one more instruction and then causes ~~the~~ a current context to be swapped out.

Claims 27-29 (Canceled)

30. (New) The execution unit of claim 1 wherein the register set comprises a first number *n* of two-ported random access memory devices, a second number *r* of effective read ports, and a third number *w* of effective write ports, where $n \geq 2$, $2 \leq r \leq n$, and $2 \leq w \leq n-1$.

31. (New) The method of claim 15 wherein storing and obtaining operands comprises storing and obtaining operands within the general purpose register comprising a first number *n* of two-ported random access memory devices, a second number *r* of effective read ports, and a third number *w* of effective write ports, where $n \geq 2$, $2 \leq r \leq n$, and $2 \leq w \leq n-1$.

32. (New) The execution unit of claim 19 wherein the general purpose register set comprises a first number *n* of two-

ported random access memory devices, a second number r of effective read ports, and a third number w of effective write ports, where $n \geq 2$, $2 \leq r \leq n$, and $2 \leq w \leq n-1$.